

# SHELL

石佛

# 一.SHELL 究竟是什么？

在计算机科学中，Shell俗称壳（用来区别于核），是指“提供使用者使用界面”的软件（命令解析器）。它类似于DOS下的command和后来的cmd.exe。它接收用户命令，然后调用相应的应用程序。（请看画图板演示）

同时它又是一种程序设计语言。作为命令语言，它交互式解释和执行用户输入的命令或者自动地解释和执行预先设定好的一连串的命令；作为程序设计语言，它定义了各种变量和参数，并提供了许多在高级语言中才具有的控制结构，包括循环和分支。

## 二.SHELL 的种类

## Bourne Shell

首个重要的标准Unix Shell是1970年底在V7 Unix(AT&T第7版)中引入的,并且以它的创始科技部基础条件平台“国家气象网络计算应用节点建设”(2004DKA50730)资助者Stephen Bourne的名字命名。Bourne shell是一个交换式的命令解释器和命令编程语言。Bourne shell可以运行作为login shell或者login shell的子shell(subshell)。只有login命令可以调用Bourne shell作为一个login shell。此时,shell先读取/etc/profile文件和\$HOME/.profile文件。/etc/profile文件为所有的用户定制环境,\$HOME/.profile文件为本用户定制环境。最后,shell会等待读取你的输入。

## C Shell

Bill Joy于20世纪80年代早期,在伯克利的加利福尼亚大学开发了C shell。它主要是为了让用户更容易的使用交互式功能,并把ALGOL风格的语法结构变成了C语言风格。它新增了命令历史、别名、文件名替换、作业控制等功能。

## Korn Shell

有很长一段时间,只有两类shell供人们选择,Bourne shell用来编程,C shell用来交互。为了改变这种状况,AT&T的bell实验室David Korn开发了Korn shell。ksh结合了所有的C shell的交互式特性,并融入了Bourne shell的语法。因此,Korn shell广受用户的欢迎。它还新增了数学计算,进程协作(coprocess)、行内编辑(inline editing)等功能。Korn Shell是一个交互式的命令解释器和命令编程语言。它符合POSIX——一个操作系统的国际标准。POSIX不是一个操作系统,而是一个目标在于应用程序的移植性的标准——在源程序一级跨越多种平台。

## Bourne Again Shell (bash)

bash是GNU计划的一部分,用来替代Bourne shell。它用于基于GNU的系统如Linux.大多数的Linux(Red Hat,Slackware,Caldera)都以bash作为缺省的shell,并且运行sh时,其实调用的是bash。

## POSIX Shell

POSIX shell是Korn shell的一个变种。当前提供POSIX shell的最大卖主是Hewlett-Packard。在HP-UX 11.0,POSIX shell就是/bin/sh,而bsh是/usr/old/bin/sh。

# 各大系统使用SHELL的概览

AIX缺省的是Korn Shell.

Solaris缺省的是Bourne shell.(这里指的是SUN-SOLARIS V10)

FreeBSD缺省的是C shell

HP-UX缺省的是POSIX shell.

Linux缺省的是Bourne Again shell.

MACOS缺省的是Bourne Again shell.

\*

ORACLE-SOLARIS V11以后默认为BASH

\*

## 三.我为什么要学SHELL ?

## 情景一：

**你是一个苦逼的DBA！！！！**  
**你每天要做很多重复的操作！！！！**  
**因此你身心俱疲！！！！**

## 情景二：

**你还是一个苦逼的DBA！！！！**

**你要做你们公司数据库的备份策略！！！！**

**无论是物理备份（RMAN）还是逻辑备份（数据泵,EXP），你都志得意满。可是突然发现，你要把他变成定时任务却是如此的困难！！！！**

## 情景三：

**又是你！对没错又是你，你这个苦逼的DBA!!!**

**你们数据库每天都会有报错，可是恰巧所有报错发生时间，都被你巧妙的错过了！**

**你要在海量的日志中去搜索那个关键的点！**

**因为，只是因为！！！！**

**你没有更好的手段，去做日志挖掘。**

## 情景四：

**好吧！我真的不想再说了，不过这次又是你！**

**你个苦逼的DBA！！！！**

**你要清理掉一些数据库中的“垃圾”！！！！**

**结果你却因为你不懂正则表达式！而让本是很简单的工作变得充满了变数和复杂不堪！**

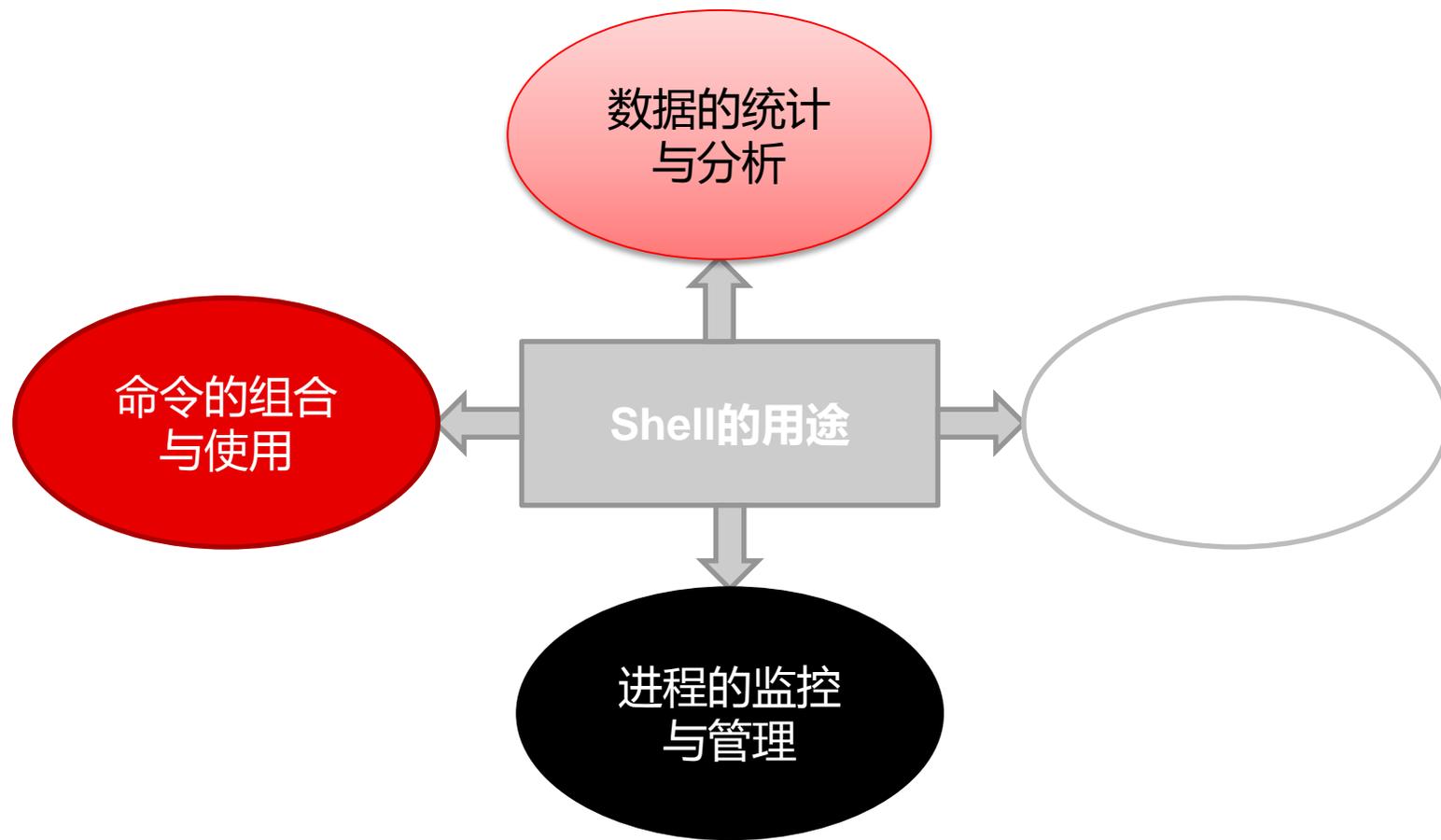
**这次看你咋办？！**

**那我们有没有好的解决方案呢？佛爷！！！！**

**有！**

**快来跟我学SHELL！！！！**

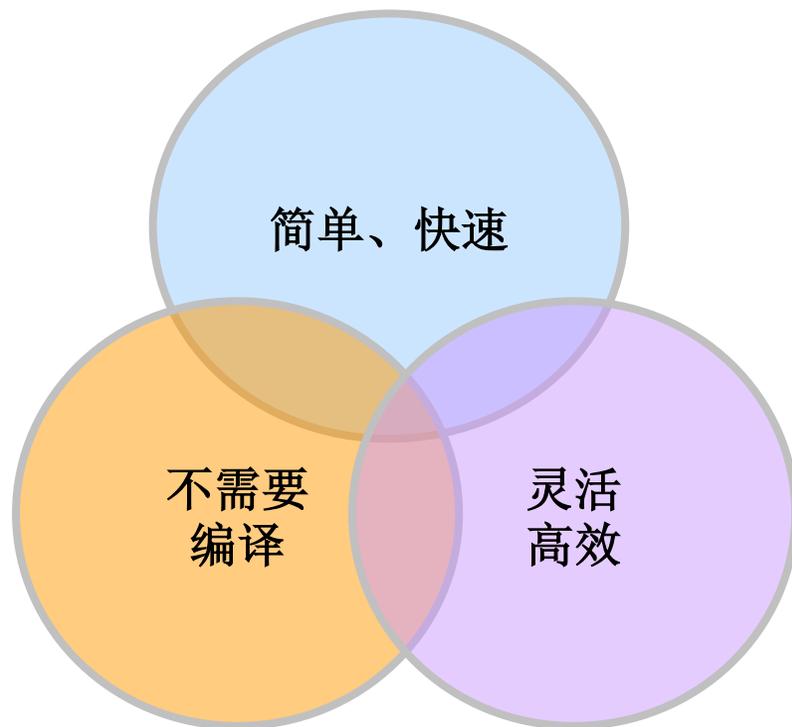
**WE WILL ROCK  
YOU！！！！！！**



# SHELL 的用途

**就让我们开始学BASH吧！！！！**

# Bash脚本优点



## 四.SHELL基础

# Bash语法

- 变量
- 数值运算
- 流程控制
- 命令
- 函数

# 变量

变量定义：`var=value`

变量引用：`$var`

# 常用变量

**\$!** 最后一个后台进程的pid

**\$\$** 当前进程的pid

**\$0** 当前程序名称

**\$n** n为1,2...n, 脚本或函数的参数

**\$#:** 所给的参数个数

**\$\*:** 所有给出的参数

**\$?** 最后一个命令的退出状态

# 流程控制

## 条件判断

- **if**
- **case**

## 循环

- **for, while**
- **break, continue**

# grep

**grep -i 忽略大小写**

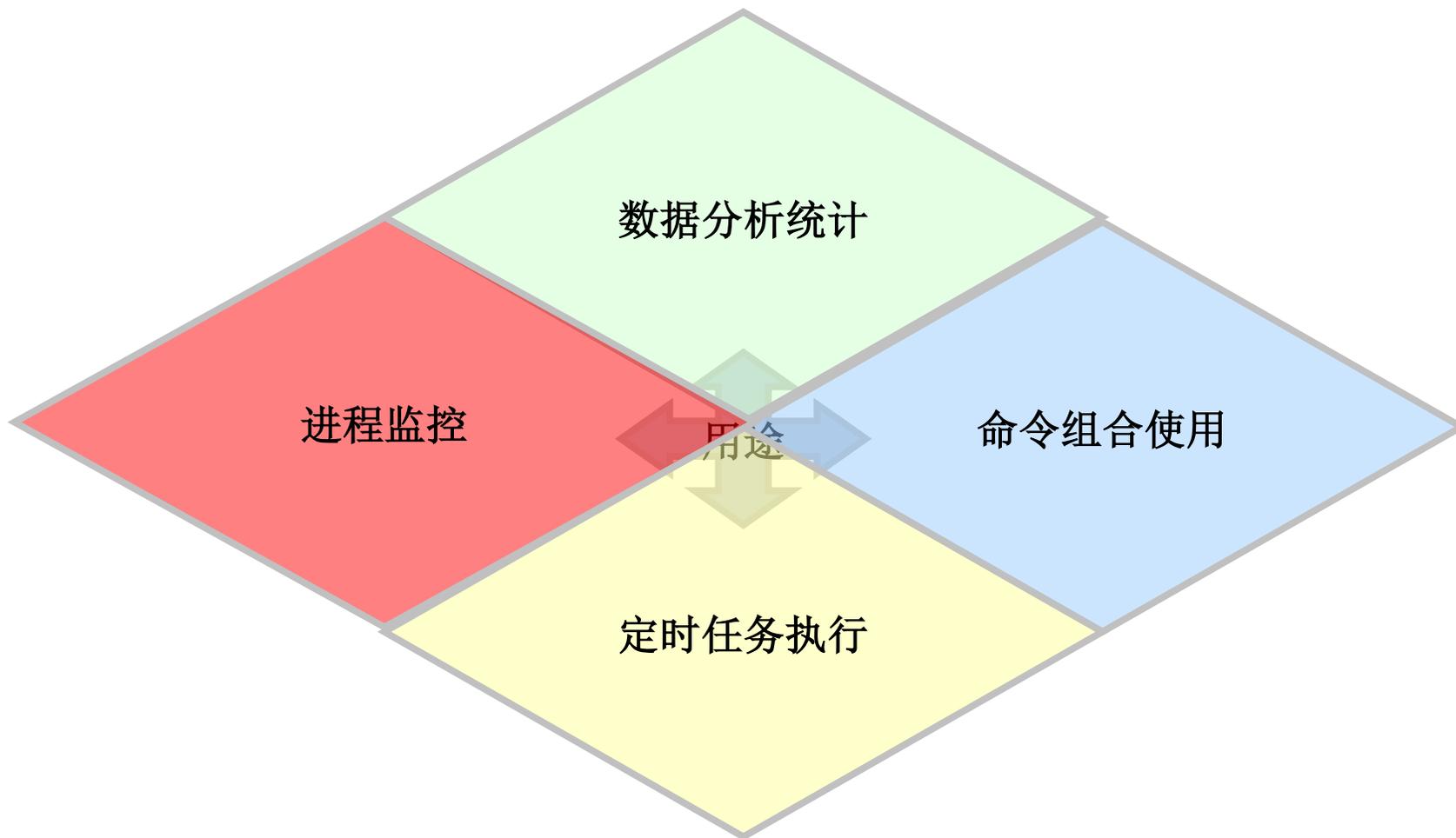
**grep -v 只显示未匹配的**

**grep -w 整个单词匹配**

**grep -n 同时打印所在行数**

**grep -E 正则匹配**

**grep --help**



# SHELL 的用途





















